parsing

DOCUMENT-IDENTIFIER: US 20020015252 A1

TITLE: Method and apparatus for a disc drive adaptive file system

---------- KWIC ---------

Detail Description Paragraph - DETX (19):
[0042] At step 620, the method 600 determines if disc drive optimization is necessary. In one aspect, disc drive optimization is defined as mapping the LBA for the files to the optimum PBA location. If the file does not require optimization, i.e., the drive is optimized, then method 600 is put on hold, e.g., to sleep until optimization is required at step 618. The method 600 is reinitialized at step 616 upon detection of a file allocation change. If optimization is required, the current PBAs locations are determined at step 622. At step 625, the method 600 parses a read command from the computer 80, and begins to read the first LBA block of the file location set by a read command. Files are then read from the media 50 according to the command. At step 630, the method 600 determines the next logical starting LBA depending upon parameters of the previous parsed command. For example, if the command was to read a media file, the next command may be to read a play list of the media file. Based on this logical determination, the method 600 moves the read/write head(s) 40 to the logical LBA starting location and reads the LBA block into memory such as buffer memory 32.

Detail Description Paragraph - DETX (22):
[0045] The optimum LBA and PBA mapping is determined at step 655. In one aspect, the drive speed is used to determine the optimum LBA to PBA mapping. For a drive that is determined too slow at step 640, the method 600 maps the LBA to a new PBA that will be in the correct physical position when needed regardless if the PBA or LBA are fragmented across the media 50. For example,

referring to FIG. 3, if the current location of the read/write head(s) 40 is
LBA zero (PBA five), and the current location of the next command data is at
LBA two (PBA seven) the seek time is greater than if the next command starting
LBAs are located at LBA one (PBA six). The seek time is increased by the fact
that head one is used to read PBA five, and that head two is reading PBA seven.
To optimize the location of the data at PBA seven, the method 600 moves the
data in PBA six to another location, scratch area, and the like, and then moves
the data from PBA seven to PBA six. The method 600 updates the physical
descriptor table to reflect the change. As the new location of the data for
the next command is physically closer and the memory was ready to accept the
data, while the data for the current command is being transmitted over the bus
42, the data for the next command is being stored in memory. Thus, regardless
of file fragmentation by locating the data in an optimum location, the data
flow into and out of the memory is effectively synchronized with the physical
location of the data.

Detail Description Paragraph - DETX (23):

[0046] For a drive that is too fast, the method 600 maps the LBAs to optimum
PBAs minimizing the number of idle revolutions, fragmentations required, or
different media 50 and head(s) 40 needed to give the drive time to be about
aligned with the proper data location. For example, referring to FIG. 3, if
the current location of the read/write head(s) 40 is LBA zero (PBA five), and
the current location of the next command is at LBA 1 (PBA six) the seek time is
less than if the next command LBAs are located at LBA two (PBA seven), as PBA
zero and PBA seven are physically further apart than PBA five and PBA six, the
drive may spin a rev in order to give the memory enough time to allow access to
the memory. Thus, the seek time is decreased by the fact that head one is used
to read PBA five and P13A six. To optimize the location of the data at PBA
six, the method 600 moves the data in PBA seven to another location such as the
reserved area, and the like, and then moves the data from PBA six to

PBA seven.
The method 600 updates the physical descriptor table to reflect the change. As
the new location of the data for the next command is physically farther the
seek time is longer, allowing the memory more time to send the data across the
bus 42, allowing the memory time to become accessible. Thus, if the seek time
is long enough, the memory has enough time to send the data across the bus 42
before reading PBA seven, allowing the data from PBA seven to be loaded into
memory without spinning a rev. If the seek time is insufficient in length,
then method 600 continues to move the file to other PBAs until the read ahead
of the next command LBAs is about synchronized. At step 660, once the optimum
PBA location is determined, then the LBAs are mapped to the PBAs for that
current command and the optimization table is updated. Subsequently, at step
670, for a host-based system, the file allocation table updates are sent to the
operating system for processing.

L Number	Hits	Search Text	DB	Time stamp
1	1281	logical adj2 address with map$4	USPAT; US-PGPUB	2003/12/09 12:47
2	331	(logical adj2 address with map$4) with block	USPAT; US-PGPUB	2003/12/09 12:36
3	136	((logical adj2 address with map$4) with block) and (RAID SAN)	USPAT; US-PGPUB	2003/12/09 12:37
4	97	((logical adj2 address with map$4) with block) and (RAID SAN)	USPAT	2003/12/09 12:50
5	1	("5438559").PN.	USPAT; US-PGPUB	2003/12/09 12:45
6	76	(logical adj block adj address LBA) with map$4 with (physical adj block adj address PBA)	USPAT; US-PGPUB	2003/12/09 13:13
7	23	((logical adj block adj address LBA) with map$4 with (physical adj block adj address PBA)) and (RAID SAN)	USPAT	2003/12/09 13:12
8	0	(((logical adj block adj address LBA) with map$4 with (physical adj block adj address PBA)) and (RAID SAN)) and pars$4	USPAT	2003/12/09 13:01
9	1	((logical adj block adj address LBA) with map$4 with (physical adj block adj address PBA)) and pars$4	USPAT; US-PGPUB	2003/12/09 13:01
10	2	((logical adj block adj address LBA) with map$4 with (physical adj block adj address PBA)) same (read write) with request	USPAT; US-PGPUB	2003/12/09 13:09
11	2171	pars$4 with (read write)	USPAT; US-PGPUB	2003/12/09 13:14
12	1	(pars$4 with (read write)) and ((logical adj block adj address LBA) with map$4 with (physical adj block adj address PBA))	USPAT; US-PGPUB	2003/12/09 13:09
13	0	(((logical adj block adj address LBA) with map$4 with (physical adj block adj address PBA)) and (RAID SAN)) and pars$4	USPAT	2003/12/09 13:12
14	99	(logical adj block adj address LBA) same map$4 same (physical adj block adj address PBA)	USPAT; US-PGPUB	2003/12/09 13:17
15	1	((logical adj block adj address LBA) same map$4 same (physical adj block adj address PBA)) and (pars$4 with (read write))	USPAT; US-PGPUB	2003/12/09 13:13
16	4312	pars$4 same (read write)	USPAT; US-PGPUB	2003/12/09 13:14
17	1	(pars$4 same (read write)) and ((logical adj block adj address LBA) same map$4 same (physical adj block adj address PBA))	USPAT; US-PGPUB	2003/12/09 13:14
18	4581	pars$4 with (request command)	USPAT; US-PGPUB	2003/12/09 13:17
19	1	(pars$4 with (request command)) and ((logical adj block adj address LBA) with map$4 with (physical adj block adj address PBA))	USPAT; US-PGPUB	2003/12/09 13:15
20	14	(pars$4 with (request command)) and (logical adj2 address with map$4)	USPAT; US-PGPUB	2003/12/09 13:15
21	16	(pars$4 same (read write)) and (logical adj2 address with map$4)	USPAT; US-PGPUB	2003/12/09 13:15
22	5980	(pars$4 decod$4) with (request command) with (read write)	USPAT; US-PGPUB	2003/12/09 13:18
23	2	((pars$4 decod$4) with (request command) with (read write)) and ((logical adj block adj address LBA) with map$4 with (physical adj block adj address PBA))	USPAT; US-PGPUB	2003/12/09 13:18
24	27434	(pars$4 decod$4) with (request command)	USPAT; US-PGPUB	2003/12/09 13:18

25	4	((pars$4 decod$4) with (request command)) and ((logical adj block adj address LBA) with map$4 with (physical adj block adj address PBA))	USPAT; US-PGPUB	2003/12/09 13:18
-	1	("6065096").PN.	USPAT; US-PGPUB	2003/12/09 12:36
-	1	(("6065096").PN.) and map$4	USPAT; US-PGPUB	2003/12/08 18:12